

OZ: 対象指向開放型分散システムアーキテクチャ

3E-9

--- 分散型デバッガの実現 ---

四反田秀樹 (松下電器) 塚本享治 (電総研)
 舟渡信彦 (シャープ) 吉江信夫 (住友電工)

1. まえがき

複数の計算機をLAN等の高速通信媒体で接続した計算機ネットワークが日常化してきている。電子メールやファイルの共有等は既に実用化され、最近では計算機ネットワーク上での分散処理へと研究が進んでいる。すでにいくつかの分散処理システムが開発されているが、プログラム開発支援環境まで整備したシステムは、ほとんど見られない。

筆者らは現在、OZと呼ぶ分散処理システムの開発を進めているが^[1]、その中で分散プログラムをテスト・デバッグするための支援環境が必須であった。本稿ではOZ開発支援の1つとして開発した分散プログラムデバッガについて、その機能および実現方法を報告する。

2. 機能

分散システムの理想的なあり方は、「計算機ネットワークを1台の大きな計算機に見せる」事である。分散プログラムデバッガにも同じ事が求められる。図1に本デバッガの主な機能を示す。本デバッガは複数計算機上の複数プロセスのデバッグのためのプロセス選択機能を有するが、それ以外は通常のソースレベルデバッガと同様の利用者インターフェイスを有する。これは、分散環境を意識しないプログラムデバッグを可能としている証でもある。

機能	コマンド
実行制御	run, cont, step, trace cancel, send, call sb, cb (ブレークポイント設定・解除)
状態調査	print, where
ソースプログラム表示	list
コンパイル	comp
プロセス選択	debug

図1 デバッガの主な機能

3. 構成と実現方法

OZシステムでは、分散処理は複数計算機上に配置されたオブジェクト間のメッセージ授受の機能で行なわれる。本分散デバッガもこの機能を利用して実現されている。図2は、デバッガというオブジェクトから、メッセージ授受という関係で結ばれているオブジェクトを示したものである。デバッガは、このオブジェクトの関係のもとで動作するが、各オブジェクトをどの計算機上に配置するかは、デバッガの動作には影響しない。

(1) ファイルオブジェクト

ソースプログラムファイルの内容を表示する時などに使用するオブジェクトである。デバッガは、

```
getlines(filename, startln, endln)
```

という要求メッセージをファイルオブジェクトに送る。ファイルオブジェクトは、内部に行番号とファイル内での位置の対応表を管理しており、応答メッセージとしてソースプログラムの指定された部分を返す。

ファイルオブジェクトは、現在はソースプログラムファイルの存在する計算機上に配置している。

(2) ターミナルオブジェクト

利用者のコマンドを受け取り、デバッガの応答を表示するためのオブジェクトである。gets、puts等のメッセージを受理する。このオブジェクトは、システムのブートプログラムの中で作成される。

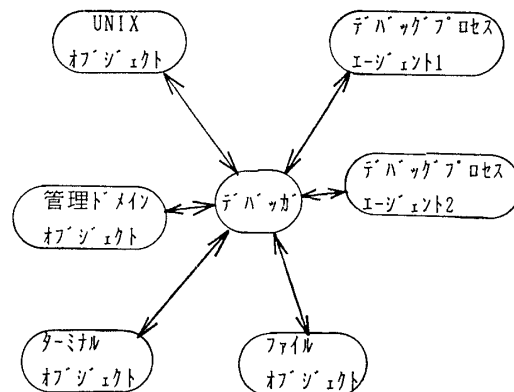


図2 デバッガのオブジェクト関係

(3) 管理ドメインオブジェクト

デバッガは利用者プログラムとソースプログラムファイルとの対応情報を取得するため、このオブジェクトとの間でメッセージを授受する。このシーケンスを図3に示す。デバッガは初期化時、管理ドメインオブジェクトから、既にロードしたプログラム(タイプオブジェクト)に関する情報を取得する([1])。また、以降ロードするたびにデバッガに通知する様、依頼する([2])。管理ドメインオブジェクトは、別オブジェクトからのデマンドロード要求等によって新たなタイプオブジェクトをロードすると、それに関する情報をデバッガに通知する([3])。デバッガはこの情報をもとに、ソースプログラムファイルを決定し、ファイルオブジェクトにメッセージを出す。

(4) UNIXオブジェクト

本システムからUNIXの機能を利用するためのオブジェクトである。UNIXの動作している計算機上に作成し、コンパイラ等UNIX上のプログラムの起動のために使用する。デバッガのcompコマンドは、このオブジェクトを使って実現されている。

(5) デバッグプロセスエージェント

デバッガの実行制御および状態調査機能を実現するためのオブジェクトであり、デバッグ対象プロセスごとに存在する。一般にプロセスはあるオブジェクトに与えられたメッセージによって作られ、複数プロセスが1つのオブジェクトを共有している。デバッグプロセスエージェントは、これらのプロセスとオブジェクトを管理する。

プロセスは、ブレークポイントに達した場合や、利用者の停止要求を受けた場合などで、デバッグ対象となる。この場合、まずデバッグプロセスエージェントが作成され、プロセスを待ち状態に置く(図4[1])。そしてデバッグプロセスエージェントをデバッグ対象プロセスリストに加える要求をデバッガに行う([2])。デバッガは、利用者からコマンドを受けると、変数内容の調査・変更、ステップ実行等の調査要求をデバッグプロセスエージェントに送る。

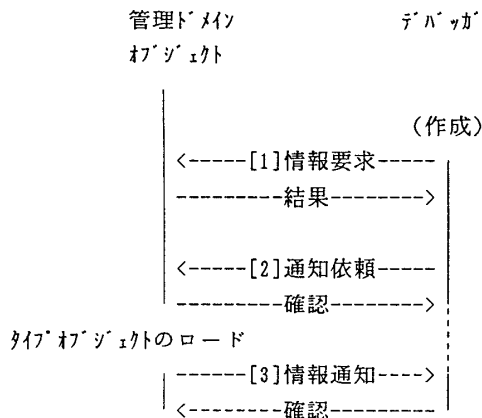


図3 管理ドメインオブジェクトとのメッセージ授受

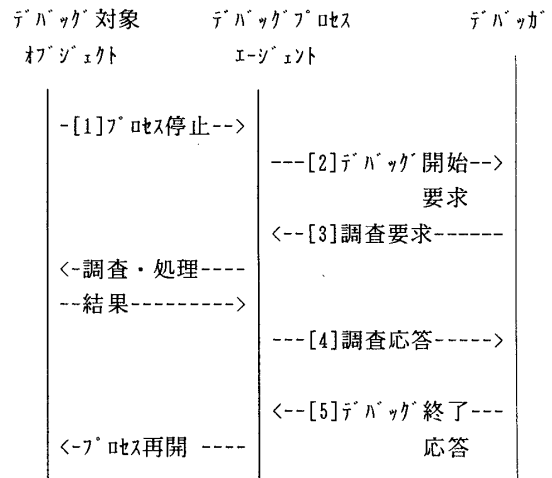


図4 デバッグプロセスエージェントとのメッセージ授受

デバッグプロセスエージェントは、このメッセージを受渡し、本来の対象プロセスあるいは対象オブジェクトの状態を調査・変更し、その結果をデバッガに返す([3][4])。

利用者がcontあるいはcancelコマンドを入力すると、デバッグ対象となっているデバッグプロセスエージェントが、デバッグ対象プロセスリストからはずされ、デバッグプロセスエージェントにデバッグ終了が通知される([5])。デバッグプロセスエージェントは消滅し、プロセスの実行が再開される。

この様に、デバッガの各機能がオブジェクト間のメッセージ授受によって実現されており、各オブジェクトの存在位置を意識する事なくデバッグが可能である。

4. まとめ

以上、分散プログラムデバッガの機能とその実現方法について述べた。これにより、分散システムの開発支援の1つが確立した。また本開発のなかで、オブジェクト内のメッセージ授受の枠組みが分散環境吸収のために有効であることが検証できた。なお、本デバッガは、複数プロセス下を想定しているが、デッドロック・実行タイミングなどの動的振舞いに関する機能はない。これらは今後の課題である。

参考文献

[1]塚本、四反田他：OZ対象指向開放型分散システムアーキテクチャ、情処第36回全国大会予稿集4G1から5